DEEPLIFT

Monday, 20th of May, 2019

Backpropagation Method : A method for decomposing the output prediction of a neural network on a specific input by **backpropagating the contributions** of all neurons in the network to every feature of the input

Reference point : the difference in output from some ‘reference’ output in terms of the difference of the input from some ‘reference’ input

let ***t*** represent some target output neuron of interest

Let ***x1,x2,...,xn*** represent some neurons in some intermediate layer or set of layers that are necessary and sufficient to compute t.

Let ***t0*** represent the reference activation of t.

Δt : difference-from-reference,that is $\Delta t = t - t0$

Contribution score : Amount of difference-from- reference in ***t*** that is attributed to or ‘blamed’ on the difference-from-reference of ***x***

Summation of C∆xi∆t = ∆t

Choose Reference point : As a guiding principle, we can ask ourselves “what am I interested in measuring differences against?”.

Example: For MNIST,we use a reference input of all zeros as this is the background of the images.

Multipliers : ***m∆x∆t = C∆x∆t / ∆x***

Chain Rule for Multipliers : ***m∆xi∆z = summation over j (m∆xi∆yj * m∆yj∆z)***

Defining Reference : references for all neurons can be found by choosing a reference input and propagating activations through the net.

Rules for assigning contribution scores

THE LINEAR RULE :

Dense and Convolutional layers (excluding nonlinearities). Let *y* be a linear function of its inputs *xi* such that ***y = b + summation over i (wixi)***. We have ***∆y = summation over i (wi∆xi).*** We define the positive and negative parts of ∆y as:

∆y+ = summation over i (1{wi∆xi > 0}wi∆xi)

∆y− = summation over i (1{wi∆xi < 0}wi∆xi)

$$C\Delta x+_i \Delta y+ = 1\{wi\Delta xi > 0\}wi\Delta x+_i$$

$$C\Delta x-_i \Delta y+ = 1\{wi\Delta xi > 0\}wi\Delta x-_i$$

$$C\Delta x+_i \Delta y- = 1\{wi\Delta xi < 0\}wi\Delta x+_i$$

$$C\Delta x-_i \Delta y- = 1\{wi\Delta xi < 0\}wi\Delta x-_i$$

$$m\Delta x+_i \Delta y+ = m\Delta x-_i \Delta y+ = 1\{wi\Delta xi > 0\}wi$$

$$m\Delta x+_i \Delta y- = m\Delta x-_i \Delta y- = 1\{wi\Delta xi < 0\}wi$$

THE RESCALE RULE *:* This rule applies to nonlinear transformations that take a single input , such as the ReLU, tanh or sigmoid operations.

$$\Delta y+ = \Delta y/\Delta x * \Delta x+ = C\Delta x+\Delta y+$$

$$\Delta y- = \Delta y/\Delta x * \Delta x- = C\Delta x-\Delta y-$$

Based on this, we get:

$$m\Delta x+\Delta y+ = m\Delta x-\Delta y- = m\Delta x\Delta y = \Delta y/\Delta x$$

Can gives misleading results, as it calculates the contribution of $\Delta x+$ ***in presence of*** $\Delta x-$ which sometimes lead to ***zero contributions of both*** $\Delta x+$ ***and*** $\Delta x-$ ***.***

Where we might prefer to use the Rescale rule?

Specifically, consider a thresholded ReLU where $\Delta y > 0$ ***iff*** $\Delta x \geq b$. If $\Delta x < b$ merely indicates noise, we would want to assign contributions of 0 to both $\Delta x+$ and$\Delta x-$ (as done by the Rescale rule) to mitigate the noise.

THE REVEAL CANCEL RULE:

By considering the *impact of the positive terms in the absence of negative terms, and the impact of negative terms in the absence of positive terms*, we alleviate some of the issues that arise from positive and negative terms canceling each other out.

$$\Delta y+ = ½ * (f(x0 + \Delta x+) - f(x0)) + ½ * (f(x0 + \Delta x- + \Delta x+) - f(x0 + \Delta x-))$$

$$\Delta y- = ½ * (f(x0 + \Delta x-) - f(x0)) + ½ * (f(x0 + \Delta x+ + \Delta x-) - f(x0 + \Delta x+))$$

$$m\Delta x+\Delta y+ = C\Delta x+y+ / \Delta x+ = \Delta y+/\Delta x+; \quad m\Delta x-\Delta y- = \Delta y-/\Delta x-$$

Problems with other back-propagation methods using gradients (like Layerwise Relevance Propagation)

1. **Saturation problem :** When value of the output function becomes constant or it saturates i.e gradient becomes zero, after a certain input value, the importance of the input features lying after that saturation point also becomes zero which is a misleading result.

However, the contribution scores calculated buy the DEEPLIFT method can be non zero even if gradient at a point is zero.

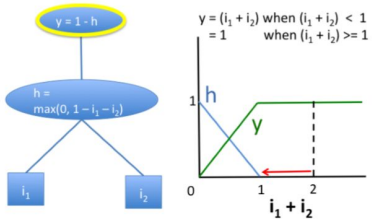

Figure 1. **Perturbation-based approaches and gradient-based approaches fail to model saturation**. Illustrated is a simple network exhibiting saturation in the signal from its inputs. At the point where $i_1 = 1$ and $i_2 = 1$, perturbing either i_1 or i_2 to 0 will not produce a change in the output. Note that the gradient of the output w.r.t the inputs is also zero when $i_1 + i_2 > 1$.

2. Thresholding problem : Sometimes there is a sudden change in the gradients such as in a following linear graph, which gives contribution scores with larger difference even in the infinitesimally small change in the input.

However, DEEPLIFT method doesn't arises such problems due to the concept of difference-from-reference. It gives continuous values in such regions.

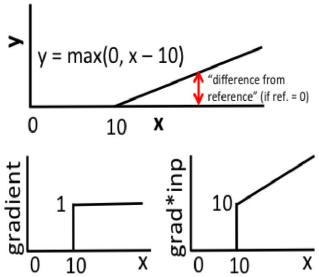

Figure 2. **Discontinuous gradients can produce misleading importance scores**. Response of a single rectified linear unit with a bias of -10. Both gradient and gradient$\times$input have a discontinuity at $x = 10$; at $x = 10 + \epsilon$, gradient$\times$input assigns a contribution of $10 + \epsilon$ to x and -10 to the bias term (ϵ is a small positive number). When $x < 10$, contributions on x and the bias term are both 0. By contrast, the difference-from-reference (red arrow, top figure) gives a continuous increase in the contribution score.

Github repository : https://github.com/kundajelab/deeplift

Work-Flow :

1. **Deeplift :** "Learning Important Features Through Propagating Activation Differences" by Shrikumar, Greenside & Kundaje
2. **Internal Influence based approach :** K. Leino, L. Li, S. Sen, A. Datta, and M. Fredrikson. Influence-Directed Explanations for Deep Convolutional Networks. In CoRR abs/1802.03788, 2018.
    a. Comparison between both
    b. Comparison with Layerwise Relevance Propagation
3. **Self Explaining Models :**

Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV)

David Alvarez-Melis, Tommi S. Jaakkola, Towards Robust Interpretability with Self-Explaining Neural Networks

Requirements:

GPU server access (CUDA) - LAPTOP

NVIDIA driver + CUDA + tensorflow + Opencv + LINUX + python - SYSTEM